NASA-TM-111513

Integrated Perception, Planning and Control for Autonomous Soil Analysis

RICHARD LEVINSON, RECOM TECHNOLOGIES
PETER ROBINSON, RECOM TECHNOLOGIES
DAVID E. THOMPSON, NASA

ARTIFICIAL INTELLIGENCE RESEARCH BRANCH
NASA AMES RESEARCH CENTER
MAIL STOP 269-2
MOFFETT FIELD, CA 94035-1000

NASA Ames Research Center
Artificial Intelligence Research Branch
Technical Report FIA-93-18
July, 1993

Reprinted from CAIA-93

Integrated Perception, Planning and Control for Autonomous Soil Analysis

Richard Levinson Recom Technologies
Peter Robinson Recom Technologies
David E. Thompson NASA

NASA Ames Research Center, AI Research Branch
MS 269-2, Moffett Field, CA, 94035
rich@ptolemy.arc.nasa.gov

Abstract

This paper describes an autonomous system that performs closed loop control of a Differential Thermal Analyzer (DTA) and a Gas Chromatograph (GC) in order to identify minerals and organics in soil samples. The system is presented as an instantiation of an integrated agent architecture designed to autonomously control scientific equipment in remote locations.

We begin by describing the motivational context and general requirements of our application. This is followed by a description of the DTA-GC problem in terms of specific requirements for integrated perception, analysis, planning and control. The next sections present the AI techniques we are applying to each of the specified requirements. Finally, we discuss the system's implementation status, and evaluate the application's effectiveness.

The original contributions of this work include a general architecture that integrates perception, analysis, planning and control for scientific experiments. For geochemists, we have contributed a new analysis instrument that integrates two previously distinct methods. The lessons imparted include issues at the integration level as well as those relating to the individual components.

AI Topic: Integrated architectures: planning, analysis, perception and control
Domain Area: Thermochemical soil analysis
Language/Tool: Common Lisp, GPIB, C
Status: The system functions as a prototype. Further development is required before completion.
Effort: 2 person years for the architecture, 2 person years for the components.
Impact: A new soil analysis technique and a prototype for autonomous control of scientific experiments.

1 Motivation

NASA missions often require that scientific experiments be performed in remote or hostile environments such as interplanetary space or planetary surfaces. Even when humans are present in these locations, they are often too busy to perform lengthy science experiments. This motivates the development of systems that perform scientific experiments autonomously. These systems must function without human assistance in uncertain and changing environments, while facing limited resources such as time and material. A higher-level executive would typically assign such an automomous system to work on a specific set of scientific goals for a specific period of time. In a planetary rover setting for instance, the system may need to complete the experiment before moving to a new location. Such scenarios motivate the requirement that the system be able to perform experiments within deadline limits. Our general problem can be described as: *Produce the best possible analysis results, given a set of scientific goals and a time limit.*

2 Application problem description

Our application addresses this requirement for "remote science" at two levels. Our most specific and immediate need is a system to autonomously control a new geochemistry instrument that combines differential thermal analysis (DTA) with gas chromatography (GC). Although the DTA and GC complement each other naturally, they have never previously been connected. This instrument is a prototype of a system that could eventually perform autonomous soil analysis on Mars. Our more general but equally important need is to develop a general architecture that can be applied to other kinds of scientific instruments. Specifically, we are also automating an instrument called a Bioreactor that controls the population density and growth rate of a microbial culture, through regulation of nutrients in a closed environment. Our application addresses the specific requirements of the DTA-GC instrument while serving as an architectural model for autonomous control of other scientific instruments.

The specific DTA-GC operational scenario can be summarized as follows: in order to identify the mineral and organic composition of an unknown soil, a soil sample is placed in the DTA oven along with an inert "reference" soil. As the oven heats up, the DTA records a voltage that indicates the *temperature difference* between the unknown soil and the reference soil. A significant difference between the sample and the reference temperatures indicates that the sample must be undergoing *exothermic* processes that produce heat or *endothermic* processes that consume heat. Some of these reactions also produce gas that is detected through pressure sensors inside the oven, and sent to the Gas Chromatograph. Proper identification of the exothermic, endothermic and gas "events" for a given soil can be used to produce hypotheses about what minerals are present in the sample. The following section motivates the use of AI techniques by further describing this problem in terms of requirements for sensory perception, data analysis, planning and control.

3 AI requirements

This application requires **sensory perception** capabilities which we define as the ability to acquire information about the external world via sensors. The system must interpret real-time DTA, GC and pressure signals from the hardware sensors. These sensors provide results in the form of voltage streams that are typically plotted graphically and then visually interpreted by humans. Since our system will be autonomous, it needs some signal processing capabilities for recognizing peak and valley features in the voltage streams.

Our system must address a form of *limited perception* since it never knows in advance which events will be encountered during the heating process. This uncertainty is compounded by signal/noise and figure/background discrimination issues. For instance, it can be difficult to discriminate between a single valley and two peaks. Thus, some heuristics are necessary to bias such decisions.

This application requires **data analysis** capabilities, which we define as any processing or reasoning over data that was acquired through sensory perception. The result of DTA-GC analysis is a set of hypotheses that postulate mineral combinations that could be contained in the sample. When a single observed event can be explained by two different minerals, multiple hypotheses will be produced. The result is a set of competing hypotheses that represent an ambiguous model of the unknown soil.

Since this is the first combined "DTA-GC" system, there are currently no experts on the analysis of this combined data. However, experts in DTA often employ a variety of heuristic knowledge when they choose between alternative hypotheses. We need to model the expert's reasoning process using a high level language so that the results will make sense to the scientists. Optimally, the scientists should also be able to develop and maintain the knowledge base themselves. This need for a high-level knowledge-based representation combined with heuristic search are the typical motivations for expert system techniques. Since a given observation may not perfectly match the generalized mineral characterization in the library, the use of probabilistic techniques is also motivated. Further, belief revision techniques are motivated by our limited perception in an uncertain world.

This application requires **planning** capabilities, which we define as the ability to select actions by performing "look ahead" or "predictive" search. Since the soil sample and its environment are unknown, an appropriate set of experiments cannot be fully designed in advance. Therefore, the system must perform online planning in order to design experiments based on knowledge gained at the remote location. Also, since competing hypotheses often exist and there is no human present, the system should autonomously take actions aimed at clarifying ambiguities.

For example, consider a case where the first run indicates only that gas evolved somewhere between the temperatures of 200 and 700 degrees. The data analysis results from that run could include two competing hypotheses: one assuming the gas was produced at 300 degrees and another assuming it happened at 600 degrees. A simple follow-up experiment could be designed to collect gas only between 100 and 400 degrees. If the gas is detected only in that smaller interval, then the second hypothesis will be thrown out.

The use of planning techniques is further motivated by the need to contend with *limited resources*. The system will not always have enough time or soil for a complete second run. Therefore, the planner must reason about resources in order to choose an appropriate experiment design strategy.

For example, a complete experiment involves heating the sample up to 1200 degrees(C) at a rate of 10 degrees/minute, thus taking about two hours. If the system has only one hour in which to clarify ambiguities that occur at 1000 degrees, there would not be enough time for a complete second run. The planner could choose a strategy that uses a much faster heating rate to "skip" the first 900 degrees, slowing down to the desired 10 degrees/minute for data collection in the critical section. When there is not even enough time or soil for a partial second run, the planner may choose between strategies that modify the current experiment and those that clarify the results without requiring the use of the hardware by simply analyzing the data differently.

The *knowledge representation* used to model these strategies should be a high-level language so that scientists can develop the strategies themselves. Additionally, the language must support heuristic search techniques, and it must be procedurally expressive enough to represent the conditional and iterative control required for encoding arbitrarily complex strategies.

The planner designs experiments based on the results of data analysis, which often contain competing hypotheses. However, those hypotheses may change at any time as unexpected exothermic, endothermic and gas events are observed. Thus the planner must operate in an *uncertain and changing* environment. In order to plan effective experiments in a changing world, the planner must be able to incorporate asynchronous sensor reports into its search process.

This application requires **real-time control** capabilities, which we define as the ability to take actions in bounded time. Our system must perform real-time control in order to react to unexpected thermal and gas events produced while heating the sample. Although the system cannot be certain in advance when these events will occur, it must respond within a bounded time of their detection. If the planner cannot produce a plan within the available time, the controller should still operate with some intelligence. Thus, it should be able to generate experiments reactively by instantiating a design strategy according to heuristics that do not involve look-ahead search.

In summary, we need to combine a mineralogical expert system with integrated sensory perception, probabilistic data analysis, planning and control. The next sections present our architecture and its components in terms of the AI techniques we applied to these requirements.

4 The general architecture

A simplified view of our architecture is illustrated in Figure 1, consisting of three elements: a hardware relay, an analysis component and a control component. The *Hardware Relay* is responsible for sending effector commands to the hardware, and receiving sensor reports from the hardware. The *Analysis* components provide sensory perception capabilities which acquire information via hardware sensors, and data analysis capabilities which reason about the sensory perception. The *Control* components provide experiment planning and real-time control capabilities.

The system accepts scientific goals and a time limit as input, includes both reactive and predictive control loops, and produces analytical results. The *reactive* control loop, indicated by the solid arrows, selects actions in bounded time by matching sensor readings against condition-action "reflex" rules. The *predictive* control loop, indicated by the dashed arrows, involves sending the analysis results to the Experiment Planner. The planner searches through a space of experiment design procedures for a useful follow-up experiment, or for modifications to the current experiment. A successful search produces a new experiment in the form of condition-action rules that are passed to the Experiment Controller.

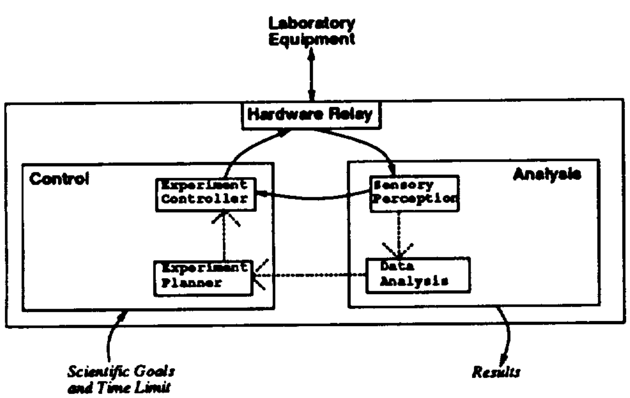

Figure 1: The general architecture

The integration of planning and control components in this architecture is based on Drummond's Entropy Reduction Engine (ERE) [2][5]. We chose the ERE approach because it has the benefit that the controller operates independently from the planner so that real-time control is not dependent on the more expensive search behavior of the planner. Our system differs from that of ERE primarily in the style of search used by the planner component. Our planner generates a task decomposition space, while their planner generates a state-space search.

5 The DTA-GC system

We now discuss how this architecture has been instantiated for the specific DTA-GC system by describing the AI techniques we applied to each component's requirements.

The Hardware Relay

The job of the hardware relay is to receive sensor readings and transmit effector commands to the hardware. The DTA-GC hardware includes a programmable Differential Thermal Analysis (DTA) oven, two Gas Chromatographs (GC), two pressure sensors, and four valves which control the gas flow between the DTA and the GC. These instruments accept over 100 distinct effector commands. In the other direction, the relay receives 9 real-time data streams from the hardware sensors including the DTA, the GC and the pressure signals.

All of the these instruments communicate with our system through a General Purpose Instrument Bus (GPIB), also called the IEEE-488 Standard. According to Caristi [4], "It is estimated that there are more than 4000 products which use the GPIB byte serial, bit parallel interface system for automatic or semiautomatic testing." This provides a large pool of laboratory equipment that could work with our system. To facilite such extension to different instruments, we have developed a general LISP/GPIB interface.

Sensory Perception

The job of the sensory perception component is to identify qualitative features (peaks and valleys) in the temperature difference, pressure and GC signals. We use a "Scale Space Filtering" technique that was developed by Witkin [13], for use in image processing domains. This technique detects peaks and valleys in a curve by convolving Gaussian filters of varying standard deviation with the input signal. As the size of the filter increases, the convolved signal becomes increasingly smoothed. Hence, the inflection points that remain after applying the larger filters correspond to the most prominent variations in the input signal. Points of inflection at varying filter scales are then grouped into scale-space contours. The input signal's first derivative is used to determine whether a given contour is a peak or a valley, and to determine a degree of belief associated with the contour.

This technique produces knowledge base predicates of the form: (Observation <Type> <Peak> <Belief>) The *type* is Exotherm, Endotherm, CO2, H20 or Organic. The *peak* indicates the temperature when the event occured. The *belief* indicates the probability that the observation really occured. This belief attribute helps to address the inherent perceptual uncertainty in our domain generated by signal/noise and figure/background discrimination issues, as well as the use of a sparse set of gaussian filters. See [10] for a main unexplained. Even for simple examples, these rules can produce many distinct explanations. This ability to automatically and systematically construct and evaluate so many alternative, yet viable, explanations could outperform the human expert who may not be so rigorous in exploring alternatives. This ability to systematically generate hypotheses about geochemical structure is similar to that of DENDRAL[3]. Our system differs significantly from DENDRAL however, since it includes closed loop control, that enables it to design and perform its own experiments. The primary output of Data Analysis is a set of *Explanations* called the *Result*.

The Experiment Planner

The job of the experiment planner is to produce an experiment that clarifies the ambiguous results of a current or previous run. A "*Clear Result*" contains only one explanation that explains all observations, but this rarely occurs. More often, the result contains multiple explanations that use different minerals to explain the same observation. Additionally, the result often contains observations that were never explained, and events that were expected but never observed. These cases represent three distinct forms of ambiguity.

The planner searches through a task decomposition space to generate a set of Experiment Control Rules (ECRs) that could clarify the given ambiguities. First, the Experiment Planner selects which ambiguities to clarify using heuristics that consider ambiguity type and resource availability. The planner then chooses among hypotheses that postulate analytical and control causes for each ambiguity. The planner then selects a *strategy* for proving the hypotheses. General strategies include: designing a second run that "skips" uninteresting temperature intervals, modifying the current run, or modifying the data analysis procedure alone. Lower-level strategies produce specific ECRs by selecting specific temperature or pressure ranges for skipping, recording or sniffing. Experiment plans that do not violate resource constraints are passed to the experiment controller.

The planner is implemented in *Propel*, a general-purpose language we designed to be procedurally expressive enough to represent real-world procedures, while maintaining the benefits of heuristic search. Propel procedures allow subgoals and other choice points to be embedded within the conditional and iterative control constructs of a LISP-like language. These procedures are used to represent our experiment design strategies. The Propel interpreter generates disjunctive experiment plans by heuristically searching through the task-decomposition space that is defined by these strategies. This behavior is similar to the "skeletal plan instantiation" technique used by MOLGEN [7], with our strategies corresponding to MOLGEN's skeletal plans. Although our representation is more procedurally expressive, the strategies in both systems encode experiment design information. In contrast with MOLGEN however, our system performs closed loop control by actually executing the experiments it designs, and analyzing their results. The expressiveness of our language is similar to McDermott's Reactive Plan Language (RPL) [11] which is an extension of Firby's RAPS language [6]. Those languages differ from ours however, since Propel was designed to generate and search for plans, while RPL and RAPS were primarily designed for reactive plan execution.

To address our *deadline management* requirements, the Planner must ensure that results are returned within the given time limit. The planner first estimates the available computation time by subtracting an initial estimate of required execution time from the given time limit. During simultaneous planning and execution, this estimate of execution time is adjusted according to the projected durations of developing experiment plans. If a plan is found within the available computation time, then it is passed to the controller for execution. Otherwise, the controller could begin execution of the default experiment, or it could reactively instantiate an experiment design strategy. This is facilitated by the Propel strategy representation which can be instantiated in bounded time using predetermined heuristics. This type of action representation, which can be used by both the planner and the controller allows for a tighter integration between planning and execution as discussed by Hanks and Firby [8], Beetz and McDermott [1], and McDermott [11].

Since the planner must operate in a changing environment, we designed a mechanism called *Dynamic Dependencies* that integrates asynchronous perception and analysis into the planner's search process. This mechanism is similar in motivation to the monitors described by Hanks and Firby in [8], but their approach is based more on decision theory than on dependency analysis. With our mechanism, the planner performs dependency analysis on the projection paths to identify external conditions on which its plans rely. The analysis component is informed about these plan assumptions so that it can notify the planner as soon as their status changes. The planner can then adjust its search control to favor plans that are based on new assumptions instead of continuing to develop plans that are based on obsolete assumptions. This technique will allow us to break the typical planning system assumption that the world does not change during the planning process. This "static world assumption" does not hold when we are planning changes to the current experiment. Performing dependency analysis on our procedurally expressive experiment strategies is a difficult task. Our approach is to extend the dependency analysis techniques used by Zabih et al. [14] and Kambhampati [9] to handle our procedurally expressive action representation, and to fit into our context of planning with asynchronous sensory perception.

6 Status

We have spent a significant amount of time building the DTA-GC instrument hardware itself, and the LISP/GPIB interface. We have also focused extensively on building up the mineral library used by the Classifier, and on the development of Propel and the Dynamic Dependency mechanism.

We now describe the implementation status of our application in more detail in terms of three progressive development levels. The *first level* represents our "baseline" functionality, by providing a reactive control loop. This level requires the operation of the sensory perception, experiment control, and data analysis components. At the *second level*, the predictive control loop is added by introducing the Experiment Planner. At this level, data analysis and experiment planning operate in sequence after the current experiment has completed. At the *third level*, all of the components operate in parallel. The static world assumption no longer holds at this level, since perception, analysis, planning and control are all being performed simultaneously.

1. **A reactive control loop.** We have demonstrated this baseline level of functionality for the reactive control loop of our system. In particular, the system can execute the default Experiment Control Rules which heats a sample slowly while monitoring the incomming DTA, GC and pressure signals. If the pressure reaches a given threshold, our system automatically reacts by evacuating the gas into the GC for analysis, and then it prepares for the next gas event. The operation of the sensory perception, data analysis and experiment control components is required at this level, so their status is described next.

   The **sensory perception** component is implemented but needs some tuning. Preliminary tests of this component were successful and are described further in [10]. We have also developed a technique called "Onset Detection". This alternative method identifies points where the curve transitions from one differential equation approximation to another. These points should indicate more accurately when the underlying chemical processes begin and end. This approach has the potential to open up a new scientific method that would identify mineral decomposition events in terms of their underlying chemical processes, instead of the weaker peak and valley descriptions currently used.

   The **experiment controller** has been implemented. We have demonstrated the ability to react to detected gas events within one second. Since the controller is a simple rule-system, it was straightforward to implement. However, the current default Experiment Control Rules are rather brittle and provide little coverage for unexpected events. Thus, we will be developing a more robust set of default ECRs through knowledge engineering efforts.

   The **data analysis** component has been implemented and produces explanations, but the rules and heuristics it uses need to be tuned through additional knowledge engineering efforts. Capturing this knowledge is necessarily slow since no one has previously performed computer analysis of simple DTA data, let alone DTA-GC data. Our mineral knowledge base has been completed and includes characterizations for over 30 classes of minerals based on over 100 experiment runs. Eventually, we intend to address issues of modeling non-linear effects of mineral combinations, and aggregate structures such as rocks which are composed of many samples, and the environment which is composed of many rocks.

2. **A serial predictive control loop.** This second level primarily involves the introduction of the Experiment Planner component and the development of better modelling and heuristic control techniques for the data analysis component. At this level, the planner can suggest follow-up runs that could produce better explanations. Since the planner still operates in sequence after data analysis at this level, the static world assumption still holds.

   The **experiment planner** component has been prototyped but needs further development. In particular, the Propel language for representing and searching through experiment strategies is implemented but the knowledge engineering of these strategies has just begun. Since the DTA-GC is a new instrument, there are no existing strategies, and our expert will first have to develop them. At this level, we also introduce deadline limits into the problem. The deadline management mechanism has been partially designed but has not been completely implemented.

3. **A parallel predictive control loop.** At this third level, all components operate in parallel, so the static world assumption no longer holds. Thus this is the first phase where the Dynamic Dependency mechanism will be required. The Dynamic Dependency technique was originally designed for the state-space search approach of the ERE reactor. We are currently re-designing it to work for Propel's task-decomposition space by extending Zabih et al's work on dependency analysis in Non-Deterministic Lisp [14].

7 Evaluation

Although we have not yet completed the system, we feel it is important to focus on a clearly defined metric that can be used to evaluate the effectiveness of our system. We have therefore designed such a metric that can be applied at each development level. We expect experiments with this metric to show that the system produces more accurate and less ambiguous results as the development level increases.

The metric characterizes the quality of the experiment results by considering how well it matches a human's analysis and how many ambiguities it contains. We evaluate how well our system identifies minerals and organics in a set of benchmark "unknown" mineral mixtures that have been provided by our scientist. A performance level of 100 percent indicates that our system produced exactly the same explanation as our human expert, with no ambiguities. From this "perfect" score, we subtract points for ambiguities in the more complete description of the sensory perception component.

The Experiment Controller

The Experiment controller is a rule-based system that matches sensory enablement conditions to GPIB effector commands. Its job is to control the laboratory equipment in real-time according to a set of Experiment Control Rules (ECRs), which are either provided by the scientist or synthesized by the Experiment-Planner.

Our controller is based on the "Reactor" and "Situated Control Rule (SCR)" elements of Drummond's ERE architecture [2] [5]. With this approach, the controller operates in a perpetual sense-act cycle, executing rules that function as quick reflexes to provide the "reactive" control capabilities of the system. In the DTA-GC system, the controller must be able to react to unexpected thermal and gas events within seconds of their detection in order to properly record and identify them.

Although many types of low-level commands can be sent to the DTA-GC instrument, we have defined three abstract operations that characterize our required experiment control behavior. These commands are: Record, Skip and Sniff. *Record* causes the oven to heat up *slowly* for some period of time during which data *will* be collected. *Skip* causes the oven to heat up *quickly* for some period of time during which data will *not* be collected. *Sniff* causes gas to be passed to one of two GCs for some period of time.

The default experiment consists of two rules that function as a set of default reflexes for the Experiment Controller. The first rule says "IF (the oven temperature is equal to *initial-temperature*) THEN Record". The second rules says: "IF (the oven pressure is greater than *pressure-threshold*) THEN Sniff". These reflexes will produce good results in cases when 2 hours are available and only one gas event occurs. More complex rules are needed for more complex experiments.

Data Analysis

In the DTA-GC system, *data analysis* corresponds to generating hypotheses that postulate mineral combinations contained in the sample. We generate hypotheses through a two step method: Bayesian Classification and Heuristic Search. The Classifier uses a Bayes tree to probabilistically match observations against events associated with known minerals in its library. The library contains knowledge of thermal and gas evolution events for over 30 primary minerals including clays, carbonates and salts.

The Classifier defines a Bayes tree for each mineral. Each child of a root *mineral node* defines a *process node* corresponding to a phase transition or chemical reaction that is caused by heating the mineral. Each of these process nodes has a terminal child node which corresponds to a specific mineral event. These *mineral event nodes* test observations for membership in a class of endotherm, exotherm, or gas events that occur within a given temperature range. The classifier uses the probabilities generated during sensory perception to assign probabilities to the terminal nodes in the Bayes trees. Using the conditional probability links from mineral-event nodes to process nodes and from process nodes to mineral nodes, a standard Bayes tree propagation algorithm [12] is used to deduce the propabilities of all non-terminal nodes. The minerals are then sorted according to their associated degrees of belief.

The Classifier produces knowledge base predicates of the form: (Match <Observation> <mineral-event> <belief>). This predicate indicates that a given observation is an instance of a particular class of mineral decomposition events. The belief attribute helps to address domain uncertainty by indicating the probability that the observation really is an instance of mineral event.

Two issues arise with the output of the Classifier. First, since the mineral events in our library may overlap, the Classifier may match a single observation to multiple mineral events, thus increasing the belief in multiple minerals based on the same piece of evidence. For example, both types of clays, montmorillonite and kaolinite, may match a single observed exotherm at 1000 degrees. The second issue is that each mineral model will only account for a subset of the observations. Thus another procedure is required to provide global explanations for the entire set of observations. In order to address these two issues, we pass the Classifier output to the Explainer, which constructs systematic explanations for the set of observations as a whole.

The Explainer is a general purpose inference engine that uses the local matches provided by the Classifier to construct *explanations* (a.k.a. *hypotheses*) for the set of observations as a whole. Each explanation contains a set of distinct mappings from each observation to a unique mineral decomposition event. This is done by reasoning about the matches provided by the Classifier. The Classifier can match a single observation to two different mineral events, or it can match a single mineral event to two different observations. Each of these cases produces disjunctive explanations. Thus, in our above example, one explanation will match the exotherm to the kaolinite decomposition event while another explanation matches it to the montmorillonite decomposition event. More disjunction is introduced to model cases where an observation is left unexplained.

The Explainer searches through this space of alternative explanations with the aid of a heuristic control function that combines multiple scoring dimensions. This heuristic is a form of Occam's Razor which prefers explanations that minimize the number of minerals used, the number of unmatched observations, and the number of unobserved events, while maximizing the combined probabilistic beliefs of the observations and the mineral events.

The Explainer currently uses two very simple hypothesis generation rules. The first rule defines a search space that matches each set of observations to a distinct set of classifications. The second rule completes the search space by allowing observations to reform of competing explanations, unmatched observations, and unobserved events that were expected. Although we currently measure our results against those of a human expert, our system could eventually outperform humans due to the systematicity and completeness of our automated approach.

We have performed some preliminary tests by running the data analysis component on a variety of mixtures. The best explanations were produced when the thermal decomposition processes of minerals in the mixture did not interact. In some of these cases, our system even suggested valid new combinations of minerals that were not hypothesized by our domain experts. On the other hand, the performance degrades when the identifying features are masked or shifted by chemical interactions between mineral processes. This issue of recognizing non-linear mineral combination effects is a focus of our future work.

8 Conclusion

We have described an architecture designed to autonomously control a new geochemistry instrument. The system functions as an instantiation of a general class of autonomous scientific instruments that integrate sensory perception, data analysis, experiment planning and experiment control. We have described how these components function and how they interact to provide autonomous control of the DTA-GC instrument.

We have developed a LISP/GPIB interface and the Propel language as general tools that could be useful for many applications. Further, the architecture will be used as a model for other intelligent instruments. In addition to these AI contributions, we have provided a contribution to soils analysis by connecting a Differential Thermal Analyzer to a Gas Chromatograph for the first time. In fact, computer assisted analysis of DTA curves is itself a new and potentially useful scientific contribution. Additionally, our use of scale-space filtering to analyze DTA curves has been published in the chemistry literature [10].

The system we have described represents a synergy between AI applications and AI techniques. Our application has stimulated the development of techniques that are useful for the integration of perception, planning and control. These techniques will in turn allow us to tackle new real-world applications that are even more ambitious.

Acknowledgements

We gratefully acknowledge the extensive collaboration of our geochemists Rocco Mancinelli and Lisa White. We also appreciate the efforts of Deepak Kulkarni, who developed our Classifier, and Kiriakos Kutulakos, who aided Deepak in the development of the sensory perception component. Thanks too to TA Instruments, who provided proprietary information about how to control their DTA oven. Finally, we thank Steve Minton and Keith Swanson for useful comments an earlier draft.

References

[1] Beetz, M., McDermott, D., 1992. Declarative Goals in Reactive Plans. in *Artificial Intelligence Planning Systems.* Edited by J. Hendler. Morgan Kaufman Publishers.

[2] Bresina, J. and Drummond, M. 1990. Integrating Planning and Reaction: A Preliminary Report. *Proceedings of the 1990 AAAI Spring Symposium Series (session on Planning in Uncertain, Unpredictable, or Changing Environments)*, Stanford, CA.

[3] Buchanan, B.B. and E.A. Feigenbaum, DENDRAL and Meta-DENDRAL: Their Application Dimension, Artificial Intelligence, 11 (1978)

[4] Caristi, A., IEEE-488: General Purpose Instrumentation Bus Manual. 1989. Acedemic Press, Inc. San Diego, Ca.

[5] Drummond, M. and Bresina, J., 1990. Anytime synthetic projection: Maximizing the probability of goal Satisfaction. In AAAI-90. Boston, MA.

[6] Firby, R.J., 1987, An investigation into reactive planning in complex domains. In Procedings of AAAI-87. Seattle, WA

[7] Friedland, P. 1979. Knowledge-Based Experiment Design in Molecular Genetics. *Ph.D. Thesis* Stanford University, Ca.

[8] Hanks, S., Firby, R.J., 1990 Issues and Architectures for Planning and Execution. *Procedings: Workshop on Innovative Approaches to Planning, Scheduling and Control*, Edited by Katia P. Sycara San Diego, CA. Morgan Kaufmann Publishers.

[9] Kambhampati, S. 1990. Mapping and Retrieval during Plan Resue: A Validation Structure Based Approach. Proceedings from AAAI '90. Boston, MA.

[10] Kulkarni, D., Kutulakos, K. and Robinson, P. Data Analysis using Scale-Space Filtering and Bayesian Probabilistic Reasoning. 1992. Computers and Chemistry, Vol. 16, No.1, Great Britain.

[11] McDermott, D. 1990. Planning Reactive Behavior: A Progress Report. *Innovative Approaches to Planning, Scheduling and Control*, Edited by Katia P. Sycara San Diego, CA. Morgan Kaufmann Publishers.

[12] Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufman, San Mateo, 144-200

[13] Witkin, A., 1987. Scale-Space Methods, in *Encyclopedia of Artificial Intelligence*, S. Shapiro, ed., Wiley, New York, 973-980

[14] Zabih, R., McAllester, D., Chapman, D., 1987. Non-Deterministic Lisp with Dependency-Directed Backtracking. *Proc. of AAAI '87*, Seattle, WA.